

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-292331

(43)Date of publication of application : 05.11.1993

(51)Int.Cl. H04N 1/419
G06F 15/66
H03M 7/46

(21)Application number : 04-074029

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 30.03.1992

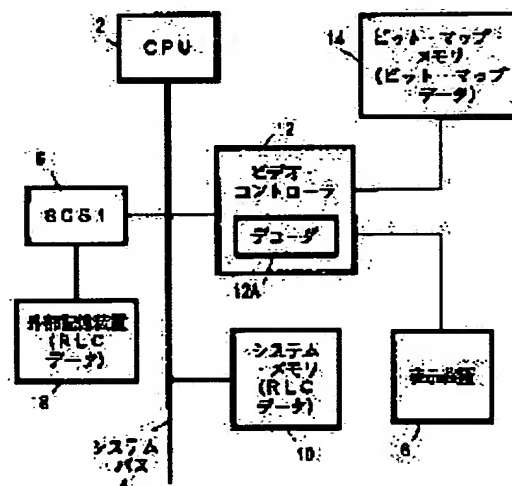
(72)Inventor : AOKI YUTAKA
AIDA YUJI
DARWIN P LACKLEY

(54) METHOD FOR DECODING RUN LENGTH CODE, VIDEO CONTROLLER, AND DATA PROCESSING SYSTEM

(57)Abstract:

PURPOSE: To improve the data writing speed to a bit map memory at the time of decoding a run length code by reducing the data writing frequency to the bit map memory by providing a skip entry.

CONSTITUTION: An external storage device 8, such as the CD-ROM, etc., is connected to a CPU 2 through a system bus 4 and an SCSI interface 6. A system memory 10 is connected to the bus and, in addition, a bit map memory 14 and a display device 16 are also connected to the bus 4 through a video controller 12. The storage device 8 stores data for display expressed in accordance with a run length code. After the CPU 2 reads the run length code data stored in the storage device 8 and holds the data in the memory 10, a video controller 12 decodes the data and writes the decoded data in a memory 14. The controller 12 reads out the data from the memory 14 and transfers the data to the display device 16 after converting the data into video signals. Therefore, the data writing speed to the memory 14 can be improved by reducing the data writing frequency to the memory 14.



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-292331

(43)公開日 平成5年(1993)11月5日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 1/419		8839-5C		
G 0 6 F 15/66	3 3 0 E	8420-5L		
H 0 3 M 7/46		8836-5J		

審査請求 有 請求項の数5(全 18 頁)

(21)出願番号 特願平4-74029

(22)出願日 平成4年(1992)3月30日

(71)出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MASCHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72)発明者 青木 豊

神奈川県大和市下鶴間1623番地14 日本ア
イ・ビー・エム株式会社 大和事業所内

(74)代理人 弁理士 頓宮 孝一 (外4名)

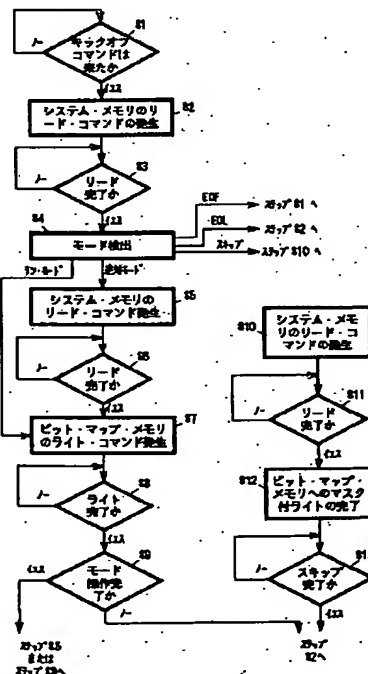
最終頁に続く

(54)【発明の名称】 ラン・レンジス・コードのデコード方法、ビデオ・コントローラ、及びデータ処理システム

(57)【要約】

【目的】 ラン・レンジス・コードをデコードしてビット・マップ・メモリへ書き込む速度を向上させる。

【構成】 デコードされたデータをビット・マップ・メモリへ書き込む際に、ラン・レンジス・コード・データのフロー中に含まれる所定のコードに応答して、データを書き込むべきビット・マップ・メモリ中の場所を所定の範囲だけ飛び越すようにして、ビット・マップ・メモリへの書き込み動作の回数を減らすようにした。



【特許請求の範囲】

【請求項1】ラン・レンジス・コードをデコードし、デコードされたデータをビット・マップ・メモリに書き込む際に、ラン・レンジス・コード・データのフロー中に含まれる所定のコードにตอบสนองして、データを書き込むべき前記ビット・マップ・メモリ中の記憶位置を所定の範囲だけ飛び越すようにした、ラン・レンジス・コードのデコード方法。

【請求項2】ラン・レンジス・コードをデコードし、デコードされたデータをビット・マップ・メモリに書き込む機能を有するビデオ・コントローラであって、ラン・レンジス・コード・データのフロー中に含まれる所定のコードにตอบสนองして、データを書き込むべき前記ビット・マップ・メモリ中の記憶位置が所定の範囲だけ飛び越されるようになっているビデオ・コントローラ。

【請求項3】前記所定のコードは、前記ビット・マップ・メモリ中の記憶位置の飛び越しの実行を指定する部分と、飛び越しの範囲を指定する部分とを有する、請求項2のビデオ・コントローラ。

【請求項4】表示装置と、ラン・レンジス・コードで表された表示用データを保持する外部記憶装置と、外部記憶装置から表示用データを読み出すCPUと、外部記憶装置から読み出された表示用データを保持する領域を有するシステム・メモリと、表示装置上の表示位置に対応した記憶場所に表示用データを保持するビット・マップ・メモリと、システム・メモリ内の表示用データをデコードし、デコードされたデータをビット・マップ・メモリに書き込むためのデコーダと、前記デコーダを含み、ビット・マップ・メモリ内の表示用データ及び表示用データの記憶場所に従って前記表示装置に映像信号を送るビデオ・コントローラと、を有するデータ処理装置であって、前記デコーダは、ラン・レンジス・コード・データのフロー中に含まれる所定のコードにตอบสนองして、データを書き込むべき前記ビット・マップ・メモリ中の記憶場所を所定の範囲だけ飛び越すようになっているデータ処理システム。

【請求項5】前記デコーダは、前記CPUにより書き込み可能なレジスタと、前記レジスタにตอบสนองして動作を開始するとともに、CPUにより読み出された表示用データの示す動作モードに従って動作するオペレーション・コントローラと、オペレーション・コントローラにตอบสนองして、CPUにより読み出された表示用データに基づいてビット・マップ・メモリに書き込むべきデータを生成するグラフィック・データ・コントローラと、オペレーション・コントローラにตอบสนองして、ビット・マップ・メモリへの書き込みアドレスを生成する物理アドレス発生器と、を有している請求項4のデータ処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ラン・レンジス・コー

ドに係り、更に詳しくは、ラン・レンジス・コードのデコードに関する。

【0002】

【従来の技術】ラン・レンジス・コード法はカラー・イメージ・データのデータ圧縮方法の1つとして知られている。ラン・レンジス・コードの一例はバイトの対から成り、第1のバイトは表示すべきカラーを指定し、第2のバイトは第1のバイトで指定したカラーが連続する表示装置上のピクセル数を指定する。カラー・イメージ・データとしてのラン・レンジス・コードはCD-ROM等の外部記憶装置からデータ処理装置内のシステム・メモリに移された後、デコードされてビット・マップ・メモリに保持される。ビット・マップ・メモリは、表示装置の表示画面上の表示位置に対応した記憶位置を備えており、ビット・マップ・メモリに書き込まれたデータは、ビデオ・コントローラの働きにより表示装置に表示される。ところで、従来のラン・レンジス・コードをデコードしてビット・マップ・メモリに書き込む方法は、必ずしも処理速度の速いものではなかった。

【0003】尚、特開昭61年第7769号公報には、イメージ・データをラン・レンジス・データに変換するための変換手段と、ラン・レンジス・データを保持するためのイメージ・メモリと、変換済みのデータをイメージ・メモリに書き込む前に1或は0の何れかをイメージ・メモリの全ての領域に書き込む手段と、イメージ・メモリのアドレス指定用のレジスタ内の値にラン・レンジス値を加算するための加算器と、を有し、イメージ・メモリに既書き込まれている初期値(11或は0の何れか)と同じデータについてはイメージ・メモリに書き込みを行わずにアドレス指定用のレジスタ内の値を更新し、初期値と反対のデータについてだけイメージ・メモリに書き込みを行うような、イメージ・メモリ書き込み制御方式が記載されている。しかしながら、この従来技術は、ラン・レンジス・コードをラン・レンジス・コードのままメモリに書き込む技術に関しており、ラン・レンジス・コードのデコードに関するものではない。また、この従来技術はモノクローム・イメージのデータにしか適用できず、カラー・イメージ・データには適用できない。

【0004】

【解決しようとする問題点】本発明の目的は、ラン・レンジス・コードをデコードしてビット・マップ・メモリへ書き込む速度を向上させることである。

【0005】

【課題を解決するための手段】前記目的を達成するため、本発明に係るラン・レンジス・コードのデコード方法は、デコードされたデータをビット・マップ・メモリに書き込む際に、ラン・レンジス・コード・データのフロー中に含まれる所定のコードにตอบสนองして、データを書き込むべきビット・マップ・メモリ中の場所を所定の範

囲だけ飛び越すようにして、ビット・マップ・メモリへの書き込み動作の回数を減らすようにした。また、本発明に係るビデオ・コントローラは、ラン・レンジス・コード・データのフロー中に含まれる所定のコードに回答して、データを書き込むべきビット・マップ・メモリ中の場所が所定の範囲だけ飛び越されるようになっている。

【0006】

【実施例】以下、本発明の実施例を図面に基づいて説明する。図1には本発明に係るデータ処理システムの一実施例の全体構成が示されている。図中、CPU2にはシステム・バス4が接続され、システム・バス4にはSS1インターフェース6を介する等してCD-ROM等の外部記憶装置8が接続されている。また、システム・バス4にはシステム・メモリ10が接続され、更に、ビデオ・コントローラ12を介してビット・マップ・メモ

リ14及びカラーCRT等のカラーの表示装置16が接続されている。外部記憶装置8にはラン・レンジス・コードに従って表された表示用データが保持され、外部記憶装置8内のラン・レンジス・コード・データはCPU2により読み出されてシステム・メモリ10内に一旦保持された後、ビデオ・コントローラ12によりデコードされ、デコードされたデータはビット・マップ・メモリ14に書き込まれるようになっている。ビット・マップ・メモリ14はビデオ・バッファの1種である。ビット・マップ・メモリ14内の記憶位置と表示装置16の表示位置とが対応関係を有している。ビデオ・コントローラ12はビット・マップ・メモリ14からデータを読み出してビデオ信号に変換した後に表示装置16に転送する。

【0007】

【表1】

第1のバイト (Lバイト 偶数アドレス)	第2のバイト (Hバイト 偶数アドレス)	定 義
00H	00H	表示ラインの終了 (EOL: End of Line)
00H	01H	フレームの終了 (EOF: End of Frame)
00H	02H	スキップ・エントリ
00H	03H - FFH	絶対モード・エントリ
01H - FFH (ピクセル数指定)	00H - FFH (カラー指定)	ラン・モード

本実施例のラン・レンジス・コードは、表1に示されるように、2バイト(1ワード)で構成される。ラン・レンジス・コードの第1のバイト(Lバイト)は同一のカラーを連続して表示させるピクセル数を指定し、第2のバイト(Hバイト)はカラーの種類を指定する。ラン・モードでは、連続する2つのバイトがラン・レンジス・コードであるとの前提に立って、第2のバイトで指定されたカラーを第1のバイト指定されたピクセル数に対応する数だけ繰り返してビット・マップ・メモリ14の記憶場所へ書き込むようビデオ・コントローラ12が動作する。絶対モードでは、絶対モード・エントリ(Lバイト: 00H、Hバイト: 03H-FFH)の後、第1のバイトも第2のバイトも単にピクセルのカラーを指定しているという前提に立って、システム・メモリ10のデータをビット・マップ・メモリ14に書き込むようビデオ・コントローラ12が動作する。

【0008】第1のバイト(Lバイト)が00Hのコードをエキストラ・コードと呼ぶことにする。表1にはエキストラ・コードの指定内容が示されている。第2のバイト(Hバイト)が00Hのエキストラ・コードは表示ラインの終了(EOL)を指定する。このコードに対しては、ビデオ・コントローラ12はビット・マップ・メモリ14の書き込み場所(アドレス・ポイント)を、現在の表示位置が表示ラインの途中であっても、次の表示ラインの先頭の表示位置に対応するアドレスまで移動させる。第2のバイトが01Hのエキストラ・コードはフレームの終了(EOF)を指定する。このコードに対しては、ビデオ・コントローラ12はビット・マップ・メモリ14のアドレス・ポイントを、次のフレームの先頭の表示位置に対応するアドレスまで移動させる。第2のバイトが03HからFFHのエキストラ・コードは絶対モードを指定する。このコードに対しては、第2のバ

トで指定されるピクセル数のデータをシステム・メモリ 10 から順次読み出してビット・マップ・メモリ 14 に書き込む。

【0009】第2のバイト(Hバイト)が02Hのエキストラ・コードをスキップ・エントリ(飛び越し指定)と呼ぶことにする。スキップ・エントリは、スキップ・エントリの後続の2バイトにより指定される範囲だけ、ビット・マップ・メモリ 14 のアドレス・ポイントを移動させる。スキップ・エントリの後続の第1のバイト(Lバイト)は水平方向に飛び越すピクセル数を指定し、第2のバイト(Hバイト)は垂直方向に飛び越す表示ライン数を指定する。スキップ・エントリにより飛び越された領域内のビット・マップ・メモリ 14 中のデータに対しては書き換え動作は行われず、それまでのデータが単に維持される。

【0010】図2には、システム・メモリ 10 内のデコード前のデータとビット・マップ・メモリ 14 内のデコード後のデータの一例が示されている。尚、この例では1ピクセル当りが8ビットのデータで表される。システム・メモリ 10 のアドレスn番地内の「03H」とアドレス(n+1)番地内の「77H」は、77Hで指定されるカラーが3ピクセル連続することを表している。従って、それはデコード後には「77H、77H、77H」となる。アドレス(n+2)番地内の「00H」とアドレス(n+3)番地内の「05H」は、エキストラ・コードの内の絶対モード・エントリである。この絶対モード・エントリは、絶対モードが5ピクセル数だけ連続することを表している。従って、絶対モード・エントリ後の5バイトのデータ「55H、AAH、33H、CCH、EEH」はビット・マップ・メモリ 14 にもそのまま書き込まれる。この際、アドレス(n+9)番地のデータは無視され、ビット・マップ・メモリ 14 内のデータには影響しない。アドレス(n+10)番地内の「03H」とアドレス(n+11)番地の「11H」は、11Hで指定されるカラーが3ピクセル数だけ連続することを表しているの、デコード後は「11H、11H、11H」となる。

【0011】図3にはビデオ・コントローラ12の内のラン・レングス・コード(RLC)のデコーダ12Aの構成が示されている。RLCデコーダ12Aの主要な構成要素はI/Oレジスタ群18、RLCオペレーション・コントローラ20、RLCグラフィック・データ・コントローラ30、及び、物理アドレス発生器40である。I/Oレジスタ群18はCPU2による書き込みが可能である。I/Oレジスタ群18中の動作開始レジスタ19の所定のビット位置(動作開始命令ビット)19Aの値が0から1に書き換えられるとデコーダ12Aは動作を開始する。即ち、CPU2からデコーダ12Aへの動作開始命令は動作開始レジスタ19の特定のビット値19Aを1(真)にすることである。この特定ビット

19Aの値はRLCオペレーション・コントローラ20内の次ステート発生器24(図4)に常に与えられている。RLCオペレーション・コントローラ20はラン・レングス・コードの第1及び第2のバイト部分を解釈してRLCデコーダ12Aの動作を決定する。RLCグラフィック・データ・コントローラ30はシステム・メモリ 10 からのデータに基づいてビット・マップ・メモリ 14 に書き込むべきデータを、動作モードに従って、生成する。物理アドレス発生器40はデコード後のデータを書き込む際のビット・マップ・メモリ 14 のアドレスを生成する。

【0012】図4にはRLCオペレーション・コントローラ20の構成が示されている。RLCオペレーション・コントローラ20はRLCオペレーション・コントロール・カウンタ22、次ステート発生器24、コマンド発生器26、コマンド・タイミング調整器27、RLCオペレーション・サブ・コントローラ28、ハンドシェイク信号タイミング調整器29、及び非同同期回路38を有している。また、RLCオペレーション・コントロール・カウンタ22は2つの同期フィードバック・ループ32及び34と、1つの非同同期フィードバック・ループ36とを有している。後続(次)のステートは現在ステートと2つの同期フィードバック・ループ32及び34からのデータとから生成される。次ステートはクロック信号の各立上りで生成される。

【0013】図5にはRLCオペレーション・コントロール・カウンタ22及び次ステート発生器24が示されている。RLCオペレーション・コントロール・カウンタ22は並列に配置された4つのD型フリップ・フロップ23-1、23-2、23-3、及び23-4から成り、デコーダ12Aの全体の動作を規定する複数のステート間を選移するステート・マシンの1種である。全てのD型フリップ・フロップ23-1乃至23-4は電源投入時にリセットされ、デコーダ12Aに動作開始命令が与えられるまで待機状態を続ける。次ステート発生器24はマルチプレクサ(MUX)25を備え、MUX25は動作開始命令ビット19Aにより駆動される。動作開始命令ビット19AによるMUX25の駆動はクロック信号のタイミングに合わせて行われる。MUX25の2つの入力4ビットの固定データ「0000」と4ビットの固定データ「0001」である。固定データ「0000」は待機ステートに対応し、固定データ「0001」は動作開始ステートに対応する。MUX25の4ビットの出力は同期フィードバック・ループ32を介して4つのD型FF23-1、23-2、23-3、及23-4の夫々に与えられる。

【0014】CPU2が動作開始命令ビット19Aに1を書き込むか否かによって、RLCオペレーション・コントロール・カウンタ22はRLCデコードのための最初のオペレーションを開始するか待機状態を継続する。

これは、動作開始命令ビット19Aの値によって、4つのD型FF23-1、23-2、23-3、及23-4の出力値をRLCデコードのための最初のオペレーションを示す値(0001)にするか、待機状態を示す値(0000)にすることにより実行される。動作開始後は、デコーダ12Aがデコード終了コードを受け取るまで、動作開始命令ビット19Aは動作中を示す値を保持する。デコーダ12Aが動作を終了する時は、RLCオペレーション・コントローラ20からリセット信号が動作開始レジスタ19に送られ、動作開始命令ビット12Aはリセットされる。この結果、RLCオペレーション・コントロール・カウンタ22は待機状態になる。CPU2によって次に動作開始命令ビット19Aが1になるまでRLCオペレーション・コントロール・カウンタ22は待機状態状態を継続する。

【0015】次ステート発生器24はRLCオペレーション・コントロール・カウンタ22の次ステートを算出する。次ステートの算出には次の3つの場合がある。第1の場合は、同期フィードバック・ループ32を用いる場合である。同期フィードバック・ループ32は次ステート発生器24とRLCオペレーション・コントロール・カウンタ22とから成っている。第1の場合では、次ステート発生器24はRLCオペレーション・コントロール・カウンタ22の保持する現ステートの情報だけから次ステートを算出する。

【0016】同期フィードバック・ループ32は、未定義のステートにRLCオペレーション・コントロール・カウンタ22が入り込んで動作が停止できない状態(ステート・ロック・アップ状態)になるのを回避するために用いられる。RLCオペレーション・コントロール・カウンタ22は4つのD型フリップ・フロップ23-1乃至23-4で構成されているので、最大16ステートを指定することができる。しかし、本実施例では13ステートしか指定していない(後に言及する表2参照)。ところが、ノイズ等の原因により、RLCオペレーション・コントロール・カウンタ22の出力が、未定義の3つのステート「1101」、「1110」、或は「1111」になる場合があり得る。そこで、RLCオペレーション・コントロール・カウンタ22の状態がこれらの3ステートの何れかに入ったときは、同期フィードバック・ループ32により、次のクロックで、「0000(待機ステート)」にセットされるようになっているのである。

【0017】ここで言う第1の場合には、待機ステートからRLCデコードの最初のオペレーションに移行する操作も含まれる。第1の場合では、次ステート発生器24に入力する動作開始命令ビット19Aの値が次ステート発生器24のステート決定に関与している点が、純粹に同期フィードバック・ループ32のみによるフィードバック操作とは異なっている。ステート・ロック・アッ

プの回避の他にも、同期フィードバック・ループ32のみで次ステートを決定する5つの場合がある。これらの5つの場合内の3つは、現ステートで行った操作と次に行う操作との間に中間ステートが設けられている場合である。具体的には、システム・メモリの読み出し操作を開始するステートに移行する場合とビット・マップ・メモリの書き込み操作を開始するステートに移行する場合である。他は後述の疑似書き込みとスキップ・モード終了検出後の動作ステート決定の場合である。

【0018】第2の場合には、同期フィードバック・ループ34を用いる場合である。同期フィードバック・ループ34はコマンド発生器26と、コマンド・タイミング調整器27と、RLCオペレーション・サブ・コントローラ28と、次ステート発生器24と、RLCオペレーション・コントロール・カウンタ22とから成っている。この第2の場合では同期フィードバック・ループ34の他に同期フィードバック・ループ32も用いられる。即ち、次ステート発生器24は、RLCオペレーション・コントロール・カウンタ22の保持する現ステートの情報とRLCオペレーション・サブ・コントローラ28の情報とを総合して次ステートを指定する。RLCオペレーション・コントロール・カウンタ22に最終的に与えられる次ステート値は同期フィードバック・ループ32に与えられる。

【0019】ここで言う第2の場合には以下の3つの場合が含まれる。まず、RLCコード中のモード情報を含むコードをシステム・メモリ10から読み出した後に指定されたモード(ラン・モード、絶対モード、スキップ・モード、ライン終了モード、フレーム終了モード、動作終了モードの何れか)に応じたステートに遷移する場合である。もう1つの場合は、ラン・モード中のビット・マップ・メモリ14への書き込みの繰り返す場合である。残りの1つの場合は、スキップ・モード中のスキップの継続をする場合である。

【0020】同期フィードバック・ループ34にコマンド発生器26とコマンド・タイミング調整器27とが含まれているのは次の理由による。即ち、RLCオペレーション・サブ・コントローラ28が出力する情報は直前に実行されたコマンドの結果に影響される。例えば、モード遷移時では、直前にモード情報を含むRLCコードが読み込まれ、読み込まれた情報はRLCオペレーション・サブ・コントローラ28内に保持され、どのモードであるかの解釈が行われる。つまり、複数のステート(この場合は2つのステート)に跨がってRLCオペレーション・サブ・コントローラ28の出力情報が決定され、また、全体の動作はクロックに同期したものである。同期フィードバック・ループ34内にコマンド発生器26とコマンド・タイミング調整器27とが含まれているのである。

【0021】第3の場合には、非同期フィードバック・ル

ープ36を用いる場合である。非同期フィードバック・ループ36はコマンド発生器26と、コマンド・タイミング調整器27と、非同期回路38と、ハンドシェイク信号タイミング調整器29と、次ステート発生器24とから成っている。非同期フィードバック・ループ36が用いられるのは、システム・メモリ10或はビット・マップ・メモリ14へのアクセス時である。非同期回路38はシステム・メモリ10及びビット・マップ・メモリ14とR L Cオペレーション・コントローラ20との間のインターフェースの役割を果たしている。この第3の場合では、非同期フィードバック・ループ36の他に同期フィードバック・ループ32も用いられている。但し、同期フィードバック・ループ34は用いられない。

【0022】次ステート発生器24はR L Cオペレーション・コントロール・カウンタ22とハンドシェイク信

号タイミング調整器29との出力から次ステートを決定し、同期フィードバック・ループ32の一部のパスを用いてR L Cオペレーション・コントロール・カウンタ22の入力にフィードバックする。この同期フィードバック・ループ32による次ステートの決定は4つのステートについて行われている。具体的には、非同期回路38に対してメモリ・アクセスを開始したステートの次のステートである。例えば、システム・メモリの読み出し命令を非同期回路38に与えた後は、タイミング調整後のハンドシェイク信号によって現ステートに止どまるか（即ち、メモリ・アクセスを続けるか）或は次ステートに進むかを決定する。

【0023】

【表2】

ステート・コード	ステート	次ステートの決定に寄与する フィードバック・ループの種類
0000	待 機	同期フィードバック・ループ 32
0001	モード検出/ メモリ・リード開始	非同期フィードバック・ループ 36 (真のメモリ・アクセス)
0010	モード検出/ メモリ・リード終了検出	同期フィードバック・ループ 34 (真のメモリ・アクセス)
0011	モード分岐	同期フィードバック・ループ 32 (真のメモリ・アクセス)
0100	絶対モード及びラン・モード でメモリ・リード開始	非同期フィードバック・ループ 36 (真のメモリ・アクセス)
0101	絶対モード及びラン・モード でメモリ・リード終了検出	非同期フィードバック・ループ 32 (真のメモリ・アクセス)
0110	絶対モード及びラン・モード でメモリ・ライト開始	非同期フィードバック・ループ 36 (真のメモリ・アクセス)
0111	絶対モード及びラン・モード でメモリ・ライト終了検出	同期フィードバック・ループ 34 (真のメモリ・アクセス)
1000	絶対モード及びラン・ モードで次動作決定	同期フィードバック・ループ 32 (真のメモリ・アクセス)
1001	スキップ・モードで メモリ・リード開始	非同期フィードバック・ループ 36 (真のメモリ・アクセス)
1010	スキップ・モードで メモリ・リード終了検出	非同期フィードバック・ループ 32 (メモリ疑似アクセス・ループ36を伴わない)
1011	スキップ・モードで メモリ疑似書込み	非同期フィードバック・ループ 32 (メモリ疑似アクセス・ループ36を伴わない)
1100	スキップ・モード 終了検出	同期フィードバック・ループ 34
1101	(未 定 義)	同期フィードバック・ループ 32
1110	(未 定 義)	同期フィードバック・ループ 32
1111	(未 定 義)	同期フィードバック・ループ 32

表2には現ステートに対して、同期フィードバック・ループ32、同期フィードバック・ループ34、及び、非同期フィードバック・ループ36の何れが用いられて次ステートが決定されるかが示されている。表2中、「0000」、「0011」、「0101」、「1000」、及び「1010」の5つのステートはメモリ・イニシャリゼーションを伴う。メモリ・イニシャリゼーションを伴うのは非同期フィードバック・ループ36を用いる4つの場合(「0000」、「0011」、及び「0101」)の他に、「1010(スキップ・モード)」のステートのように、疑似ビット・マップ・メモリ書き込み命令を発生し、非同期回路38からの応答を

利用しない場合も含まれている。

【0024】図6にはRLCオペレーション・サブ・コントローラ28の構造が示されている。RLCオペレーション・サブ・コントローラ28はモード検出器72、減算ピクセル数決定回路74、ライト・リピート・カウンタ76、及び、スキップ・リピート・カウンタ78を有している。モード検出器72の出力は次ステート発生器24に与えられる他、減算ピクセル数決定回路74にも与えられる。ラン・モード及び絶対モードにおいて、ビット・マップ・メモリ14への書き込みの際のライト・データのビット幅がライト・リピート・カウンタ76の値に従って決まるようになっている。また、ライト・

リピート・カウンタ76の値は、次ステートを決定するために、次ステート発生器24にもフィードバックされる。ライト・リピート・カウンタ76及びスキップ・リピート・カウンタ78の出力は減算ピクセル数決定回路74に与えられる。減算ピクセル数決定回路74は、モード検出器72の出力に応じて、1ピクセル、2ピクセル或は8ピクセルの何れかの値を、ライト・リピート・カウンタ76及びスキップ・リピート・カウンタ78から与えられたピクセル数から減じ、得られた結果をマルチプレクサ75及び77を介してにライト・リピート・カウンタ76及びスキップ・リピート・カウンタ78の夫々に与える。

【0025】ラン・モードにおいて、ビット・マップ・メモリ14に1度に書き込まれるデータのビット幅は、ライト・リピート・カウンタ76の値に応じて、以下のようにして決定される。ビット・マップ・メモリ14への書き込みデータはRLCグラフィック・データ・コントローラ30(図3)から供給されるが、書き込みデータ用バスのビット幅(バス幅)は64ビットである。従って、1ピクセルについて8ビットのデータを使用する場合は、8ピクセル分のデータを1度に書き込むことが可能である。ライト・リピート・カウンタ76は書き込むべきピクセル数に関する情報を保持している。ライト・リピート・カウンタ76の値が、ビット・マップ・メモリ14に書き込むべきピクセル数が16以上であることを示している場合は、バス幅の最大ビット即ち64ビット(8ピクセル)幅の書き込みを行う。ライト・リピート・カウンタ76の値及び物理アドレス発生器40の値によっては、ビット・マップ・メモリ14へのデータの書き込みのビット幅が2ピクセル(16ビット)或は1ピクセル(8ビット)になる場合がある。絶対モードでも、書き込みビット幅が1ピクセル又は2ピクセルの何れになるかの選択が行われる。また、ライト・リピート・カウンタ76の値が1以上ならば、ラン・モード或は絶対モードの書き込み繰り返しのステートであることが判断される。また、ライト・リピート・カウンタ76の値が0ならば、次のRLCコードを読み出すためのステートへ遷移すべきことが判断される。

【0026】スキップ・リピート・カウンタ78はスキップ・モードで用いられる。スキップ・モードにおいて、RLCオペレーション・サブ・コントローラ28は、スキップ・モード中のビット・マップ・メモリ14への書き込みは疑似命令であることをコマンド発生器26に対して伝える。この結果、非同期回路28に対してメモリ・アクセス・コマンドは伝えられない。具体的には、AND回路(図示せず)の一方の入力に書き込み命令を入力させ、他方の入力にマスク・コマンドを入力させ、出力を非同期回路28に与えている。物理アドレス発生器40はラン・モードの場合と同様にしてアドレスを増加し続ける。このとき、スキップ・リピート・カウ

ンタ78は、ラン・モードにおけるライト・リピート・カウンタ76と同様な動作を行う。疑似書き込みは非同期フィードバック・ループ36からのハンドシェイク信号を常にハンドシェイク終了状態にして2つのステートを使用している。2つのステートとは疑似書き込みステート(「1011」と書き込み終了ステート(「1100」)である。物理アドレス発生器40を増加させるためのクロック発生が必要であるから2ステートを使用している。

【0027】スキップ・リピート・カウンタ76の値は次ステートを決定するために次ステート発生器24にフィードバックされる。即ち、スキップ・リピート・カウンタ78の内容が1以上であればスキップのためのステートにすべきことが判断される。スキップ・リピート・カウンタ78の値が0であれば次のRLCコードの読み出しのためのステートに遷移すべきことが判断される。以上述べたことを簡単にまとめると、システム・メモリ10からRLCコードのスキップ・コードが読み出されると、スキップ・リピート・カウンタ78にロードされ、疑似命令終了タイミングでスキップ・リピート・カウンタ78の内容が更新される。

【0028】コマンド発生器26はシステム・メモリ10及びビット・マップ・メモリ14へのアクセス信号を、RLCオペレーション・コントロール・カウンタ22の内容に従って、発生する。また、コマンド発生器26はライト・リピート・カウンタ76及びスキップ・リピート・カウンタ78に対するカウンタ・ロード信号を発生する。コマンド発生器26がメモリ・アクセス信号を発生するのはRLCオペレーション・コントロール・カウンタ22が示すステートがRLCコードの読み出しを開始するステート、ラン・モード或は絶対モードにおいてビット・マップ・メモリ14への書き込みを開始するステートである。スキップ・モードでは疑似書き込み命令が発生するが、これは非同期回路38に与える前にマスクされる。ライト・リピート・カウンタ76やスキップ・リピート・カウンタ8に対するカウンタ・ロード信号はRLCコードの繰り返し情報のリード完了のタイミングである。

【0029】図7はRLCオペレーション・コントロール・カウンタ22の動作を説明するためのフロー・チャートである。動作開始レジスタ19の動作開始命令ビット19Aに1が書き込まれると、別言すると、ラン・レンジス・コード(RLC)の始まりを示すキックオフ・コマンドがシステム・メモリ10から読み取られると(ステップS1)、コマンド発生器26はシステム・メモリ10に対する読み出しコマンドを発生する(ステップS2)。非同期フィードバック・ループ36が働いて、リード操作の完了が検出されると(ステップS3)、RLCオペレーション・サブ・コントローラ28はどのモードの操作を行うべきかを決定する(ステップ

10

20

30

40

50

S4)。フレーム終了を指示するコード(EOF)の場合は、処理はステップS1に移る。ライン終了を指示するコード(EOL)の場合は、処理はステップS2に移る。絶対モードを指示するコードの場合は、コマンド発生器26はシステム・メモリ10のリード・コマンドを発生する(ステップS5)。

【0030】システム・メモリ10の読み出しアドレスは、デコーダ12AのI/Oレジスタ群18内のリード・スタート・アドレス・レジスタ82(図3)に予め格納されている。EOF検出時は動作を終了するため処理をステップS1に戻す。この際、動作開始命令ビット19Aはリセットされる。EOL検出時は、現在のライト・スタート・アドレスにオフセット値を加算することにより、ライト・スタート・アドレスを次の表示ラインの先頭の表示位置に対応するアドレス値にする。ライト・スタート・アドレスはライト・スタート・アドレス・レジスタ84(図3)に予め格納されている。オフセット値が加算されたことはビット・マップ・メモリ・アドレス発生器40B(図11)のビット・マップ・アドレス・カウンタ66の値にも反映される。EOL毎にオフセット値は積算され、ライト・スタート・アドレス・レジスタ84はビット・マップ・メモリ14に書き込みをしている表示ラインの先頭アドレスを常に保持している。絶対モードのRLCコードについては、システム・メモリ10の読み出しにより書き込みカラー・データを得る。リード・アドレスはリード・スタート・アドレス・レジスタ82の内容に2を加えた値である。この加算操作はデコーダ12Aが自動的に行う。

【0031】システム・メモリ10のリード操作が完了すると(ステップS6)、コマンド発生器26はビット・マップ・メモリ14への書き込みコマンドを発生する(ステップS7)。ビット・マップ・メモリ14への書き込み操作が終了したら(ステップS8)、RLCオペレーション・サブ・コントローラ28からの情報を基に次ステート発生器24は実行中のモードを終了するか否かを判断する(ステップS9)。実行中のモードを終了すると判断したときは、処理をステップS2に移す。実行中のモードを終了しないと判断したときは、実行中のモードがラン・モードのときは、処理をステップS8に移し、実行中のモードが絶対モードのときは、処理をステップS5またはステップS8に移す(ステップS9)。処理をステップS5に移すかステップS8に移すかは、物理アドレス発生器40(図3)からRLCオペレーション・サブ・コントローラ28(図4)に送られたアドレスの下位ビットに基づいて判断される。

【0032】デコーダ12Aとビット・マップ・メモリ14とは64ビット幅のバスで結ばれている。絶対モードにおいて、奇数バイト・アドレスからビット・マップ・メモリ14への書き込みが始まり、且つ、8バイト(64ビット)毎のアドレス・バウンダリを跨いでデー

タを書き込むまでにアドレスが増加されたときは、1回の1ワード(16ビット即ち2ピクセル)の書き込み操作が2回の1バイト(8ビット即ち1ピクセル)の書き込み操作に分割されることにより、アドレス・バウンダリを跨がる書き込み操作に対応している。書き込み操作がアドレス・バウンダリを跨がることになるか否かは、前述のように、次に書き込みを行うアドレスを保持している物理アドレス発生器40のアドレスの下位の数ビットを参照して行う。物理アドレス発生器40のアドレスの下位の数ビットを参照した結果は、クロス・バウンダリの有無を示す1ビットの情報となり、RLCオペレーション・サブ・コントローラ28から次ステート発生器24に送られ、次ステートの判断に利用される。こうしたクロス・バウンダリに関する処理はラン・モードでも発生し、絶対モードと同じハードウェアで実行される。

【0033】ステップS4においてラン・モードを実行すべきと判断したときは、処理はステップS7に移る。ステップS2ではシステム・メモリ10から2バイトのデータが読み込まれ、その内の1バイトはラン・モードの指定及び1回のラン・モードの連続ピクセル数を指示し、他の1バイトは書き込みカラーを指示する。書き込みカラーを指定する8ビットのデータはRLCグラフィック・コントローラ30(図3)に取り込まれ、図8に示されるように、64ビットに拡張され、ビット・マップ・メモリ14へのデータ・バス上に乗る。データ拡張操作はステップS7に至る前に終了する。

【0034】ステップS4においてスキップ・エントリを実行すべきと判断したときは、処理はステップS10に移る。ステップS10では、コマンド発生器26がシステム・メモリ10へのリード・コマンドを発生する。非同期フィードバック・ループ36の働きによりリード操作の終了が判断されると(ステップS11)、RLCオペレーション・サブ・コントローラ28の指示に従って、コマンド発生器26が働き、ビット・マップ・メモリ14への書き込み操作がマスクされる(ステップS12)。このマスク操作は非同期フィードバック・ループ36からの応答を待たず行われる。ステップS13ではビット・マップ・メモリ14へのアドレス・ポイントの飛び越し(スキップ操作)が終了したか否かがRLCオペレーション・サブ・コントローラ28により判断される。スキップ操作が終了していないときは、処理はステップS12に戻る。スキップ操作が終了しているときは、処理はステップS2に戻る。

【0035】ステップS10のシステム・メモリ10の読み込みは、スキップの垂直方向及び水平方向の範囲を得るためである。スキップの垂直方向の範囲は表示ライン数で表し、水平方向の範囲はピクセル数で表される。これらのデータはRLCオペレーション・サブ・コントローラ28に取り込まれる。図6に示されるように、RLCオペレーション・サブ・コントローラ28のスキッ

プ・リピート・カウンタ78は垂直スキップ・リピート・カウンタ78Aとを水平スキップ・リピート・カウンタ78Bとを有している。ステップS12とステップS13においては、ビット・マップ・メモリ14への書き込み信号がアクティブにならないような疑似メモリ書き込み状態が繰り返される。ライト・イネーブル信号をマスクしていることを除けば、このときの動作はラン・モードのときの動作と同じである。但し、ビット・マップ・メモリ14からの応答を待たないので、ラン・モードのときの動作に比べて12倍程度の高速で疑似ライト・オペレーションが繰り返される。

【0036】スキップ・モードにおける疑似ライト・オペレーションがこのような高速になるのは、ステートの遷移がクロック信号(50MHz即ち20ナノ秒)にタイミングを合わせて行われるのに対して、ビット・マップ・メモリ14のアクセスには平均240ナノ秒を要するためである。疑似ライト・オペレーションの度に水平スキップ・リピート・カウンタ78Bの値は8ピクセル、2ピクセル、或は1ピクセルの何れかが減じられる。水平スキップ・リピート・カウンタ78Bの値が0になると、垂直スキップ・リピート・カウンタ78Aの値が1だけ減じられるとともに、1表示ライン分のピクセル数、例えば640が水平スキップ・リピート・カウンタ78Bにロードされ、疑似ライト・オペレーションが繰り返される。こうした繰り返し操作は垂直スキップ・リピート・カウンタ78A及び水平スキップ・リピート・カウンタ78Bの内容がともに0になるまで継続する。

【0037】疑似ライト・オペレーション中はビット・マップ・メモリ14へのライト・イネーブル信号はマスクされるが、物理アドレス発生器40のビット・マップ・メモリ14のアドレス値を保持するビット・マップ・アドレス・カウンタ66(図11)の値は、水平スキップ・リピート・カウンタ78Bの値が減じられた数に応じて更新される。例えば、水平スキップ・リピート・カウンタ78Bの値が8だけ減じられると物理アドレス発生器40のアドレスは8だけ加算される。垂直スキップ・リピート・カウンタ78A及び水平スキップ・リピート・カウンタ78Bの内容がともに0になると、RLCオペレーション・サブ・コントローラ28はスキップ終了を示す信号をアクティブにし、この信号を次ステート発生器24に与える。この結果、次ステート発生器24の判断により処理はステップS2に戻る。スキップ・モードでの動作を疑似ライト状態の動作としてラン・モードの動作と共通する部分を多くしたので、スキップ・モードの動作を実行する回路とラン・モードの動作を実行する回路とは共通する部分が多くなり、ハードウェア構成の簡素化が図れる。しかも、スキップ・モードを利用する場合のメモリの書き込み操作の高速化については何等阻害されることがない。

【0038】ここで、以上をまとめると、CPU2が動作開始レジスタ19の動作開始命令ビット19Aを真にすると、デコーダ12Aが動作を開始する。先ず、物理アドレス発生器40のシステム・アドレス・カウンタ62に予めロードされていたアドレス値に基づいてRLCコードの最初の2バイトをシステム・メモリ10から読み出す。読み出した2バイトには、実行すべきモードの種類を指定する情報が含まれているので、指定されたモードに応じたステートへと処理が分岐される。指定されたモードによっては、システム・メモリ10を再度読み出す必要がある。システム・メモリ10からの読み出しはワード単位(2バイト)で行われるのに対応して、システム・メモリ10の読み出しが終了する度に物理アドレス発生器40のシステム・アドレス・カウンタ62の内容が2だけ増加される。

【0039】また、ビット・マップ・メモリ14への書き込みはラン・モード及び絶対モードで行われる。ラン・モードでは、残りの書き込みピクセル数に応じて64ビット単位、16ビット単位、或は8ビット単位の何れかの書き込みを行う。64ビット幅のバスにより16ビット単位或は8ビット単位の書き込みを行うのは通常のDRAMに備えられているライト・パー・ビット機能を用いる。絶対モードでは、書き込みがアドレス・バウンダリを跨ぐか否かに依って、16ビット単位か8ビット単位の書き込みを行う。ビット・マップ・メモリ14への書き込みアドレスは、物理アドレス発生器40のビット・マップ・メモリ・アドレス・カウンタ66(図11)によって指定される。このカウンタ66はバイト・アドレスを指し、書き込み操作終了時に書き込んだビット数に応じて8、2、或は1が加算される。スキップ・オペレーションはラン・モードの動作と基本的には同様であるが、ライト・イネーブル信号をマスクしている点がラン・モードの動作と異なる。EOFのみがデコーダ12Aの動作を終了させる。

【0040】図9にはRLCグラフィック・データ・コントローラ30の構成が示されている。RLCグラフィック・データ・コントローラ30はシステム・メモリ10からデータを読み出し、ビット・マップ・メモリ14に書き込むための準備を行う。ラン・モードのときは、読み出したデータは、システム・メモリ10からの次のデータの読み出しまで、ラッチ52が保持する。データ・ローテータ54は、システム・メモリ10からのnビット幅のデータをビット・マップ・メモリ14中にmビット幅で書き込むために用いられる。実行中のモード及び物理アドレス発生器40の発生したアドレスに従って、RLCオペレーション・コントローラ20がデータ・ローテータ54を操作する。データ拡張器56はデータをnビット幅からmビット幅に拡張する。

【0041】ラン・モードでは、データ・ローテータ54は動作せず、1ピクセル当り8ビットのデータが64ビ

ットに拡張されるだけである(図8参照)。絶対モードでは、奇数アドレスに16ビット・データ(2ピクセル)を書き込むか偶数アドレスに16ビット・データ(2ピクセル)を書き込むかによって、8ビット・データをローテイトするかどうかが決まる。16ビット・データを書き込むのは、絶対モードであることが判明したら、システム・メモリ10の読み出しをもう1度行って2ピクセル分のデータ(ワード・データ)をRLCグラフィック・データ・コントローラ30に取り込むことになっているからである。絶対モードではデータ・ローテ

【0042】図10及び図11には物理アドレス発生器40の互いに異なる部分の構成が示されている。物理アドレス発生器40はシステム・メモリ・アドレス発生器40A(図10)とビット・マップ・メモリ・アドレス発生器40B(図11)とから成り、システム・メモリ・アドレス発生器40Aはシステム・メモリ10のアドレスを発生し、ビット・マップ・メモリ・アドレス発生器40Bはビット・マップ・メモリ14のアドレスを発生する。

【0043】図10において、システム・メモリ・アドレス発生器40Aは、マルチプレクサ61、システム・アドレス・カウンタ62、及び加算器63を含んでいる。加算器63は現在アドレスに常に2バイトを加算する。システム・アドレス・カウンタ62は複数の並列のD型フリップ・フロップから構成されている。システム・メモリ・アドレス発生器40Aで用いるスタート・アドレスは、デコーダ12AのI/Oレジスタ群18内のリード・スタート・アドレス・レジスタ82(図4)内にCPU2により予めロードされている。リード・スタート・アドレス・レジスタ82に書き込みがあったときは、MUX61はレジスタ82内のスタート・アドレスを選択し、システム・アドレス・カウンタ62にもレジスタ82の書き込みの終了のタイミングでスタート・アドレスがロードされる。システム・メモリ・アドレスは常にバイト・アドレスであり、システム・メモリ読み出し終了のタイミングでシステム・アドレス・カウンタ62はクロックを与えられる。MUX61が加算器63の出力を選択しているときに、システム・アドレス・カウンタ62にクロックが与えられると、システム・アドレス・カウンタ62の内容に2が加算される。

【0044】図11のビット・マップ・メモリ・アドレス発生器40Bは、マルチプレクサ65、ビット・マップ・アドレス・カウンタ66、加算器67、及びマスク信号発生器68を含んでいる。加算器67は現在アドレスに常にjビットを加算する。jビットの値はRLCオ

ペレーション・サブ・コントローラ28からの指示により特定される。バイト・アクセスを行うときはj=1である。ワード・アクセスを行うときはj=2である。ラン・モードで64ビット(全バス幅)の書き込みを行うときはj=8である。スキップ・モードで64ビットの疑似書き込みを行うときもj=8である。ビット・マップ・メモリ・アドレス発生器40Bのスタート・アドレスはライト・スタート・アドレス・レジスタ84(図4)にCPU2により予めロードされている。レジスタ84に書き込みがあったときは、その書き込みの終了のタイミングでビット・マップ・アドレス・カウンタ66にもスタート・アドレスがロードされる。

【0045】マスク信号発生器68は64ビットのバス幅の一部をマスクするための信号(マスク信号)を発生する。マスク信号は絶対モードにおける8ビット或は16ビットのデータの書き込み時に使用される。マスク信号は64ビットのバス幅の内の書き込みを有効とするデータ・ビット位置を書き込み可能とし、その他のデータ・ビット位置を書き込み不可能にするような64ビットの信号である。マスク信号は、RLCグラフィック・データ・コントローラ30の発生する書き込みデータとタイミング・マルチプレクスされ、ビット・マップ・メモリ14を構成するDRAMのデータ・ラインに与えられる。DRAMのライト・パー・ビットの機能により書き込み不可のビットが書き込みマスクされる。尚、マスク信号発生器68の発生するマスク信号はスキップ・モードにおけるライト・イネーブル信号のマスクとは関係がない。

【0046】図12に示されるように、ビット・マップ・メモリ・アドレス発生器40BとRLCオペレーション・サブ・コントローラ28との間にはフィードバック・ループ72が設けられている。ビット・マップ・メモリ14のアドレスの下位ビットはRLCオペレーション・サブ・コントローラ28に送られた後、加算器67の動作を決定する。絶対モードにおいて、書き込みピクセルが残り2ピクセル(16ビット)以上であり、現在アドレスの下位ビットが「111」であるときは、次のビット・マップ・メモリ14の書き込みはバイト単位の書き込みを2回繰り返す、64ビットのバス幅に起因するアドレス・バウンダリを処理するようになっている。この処理のためにフィードバック・ループ72が用いられる。

【0047】このような本実施例によれば、ラン・レンジ・コードのデコードに際してスキップ・エントリを新たに設けて、ビット・マップ・メモリ14への書き込み頻度を減らすようにしたので、ラン・レンジ・コードをデコードしてビット・マップ・メモリ14へ書き込む速度を向上させることができる。また、書き換える必要のない範囲については、従来は複数のラン・レンジ・コードで表していたデータを1つのスキップ・エント

りに置き換えることが可能なので、スキップ・エントリをラン・レンジス・コードに含ませることにより、データの圧縮率を向上させることができる。

【0048】また、従来はソフトウェアによりラン・レンジス・コードをデコードすることが通常であったが、前記実施例ではハードウェアによりはラン・レンジス・コードをデコードしているので、この点でもデコードが速くなり、CPU2の負担も軽くなる。

【0049】尚、前記実施例では、ハードウェアによりはラン・レンジス・コードをデコードしていたが、ソフトウェアによりデコードするものであってもよい。また、ラン・レンジス・コードは2バイトの単位であるとして説明したが、2バイト以外の単位のコードであってもよい。また、前記スキップ・エントリは、飛び越し範囲を水平方向の飛び越し量及び垂直方向の飛び越し量で指定していたが、水平方向の飛び越し量及び垂直方向の飛び越し量の何れか一方のみを指定するようなコードであつてもよい。また、前記実施例では、水平方向の飛び越し量はピクセル数に対応し、垂直方向の飛び越し量は表示ライン数に対応したが、このような対応関係以外の指定方法であってもよい。

【0050】

【発明の効果】上述のように本発明によれば、ラン・レンジス・コードをデコードしてビット・マップ・メモリへ書き込む速度を向上させることができる。

【図面の簡単な説明】

【図1】本発明に係るデータ処理システムの一実施例の全体構成を示すブロック図である。

【図2】前記実施例におけるラン・レンジス・コードのエキストラ・コードの一例を示す図表である。

【図3】前記実施例のRLCデコーダの構成を示すブロック図である。

【図4】RLCデコーダ中のRLCオペレーション・コントローラの構成を示すブロック図である。

【図5】RLCデコーダ中のRLCオペレーション・コントロール・カウンタの構成を示すブロック図である。

【図6】RLCデコーダ中のRLCオペレーション・サブ・コントローラの構成を示すブロック図である。

【図7】RLCオペレーション・コントローラの動作の説明のための流れ図である。

【図8】RLCデコーダ中のRLCグラフィック・データ・コントローラにおけるデータ拡張に関する構成を示

すブロック図である。

【図9】RLCデコーダ中のRLCグラフィック・データ・コントローラの全体構成を示すブロック図である。

【図10】前記物理アドレス発生器のシステム・メモリ・アドレス発生器を示すブロック図である。

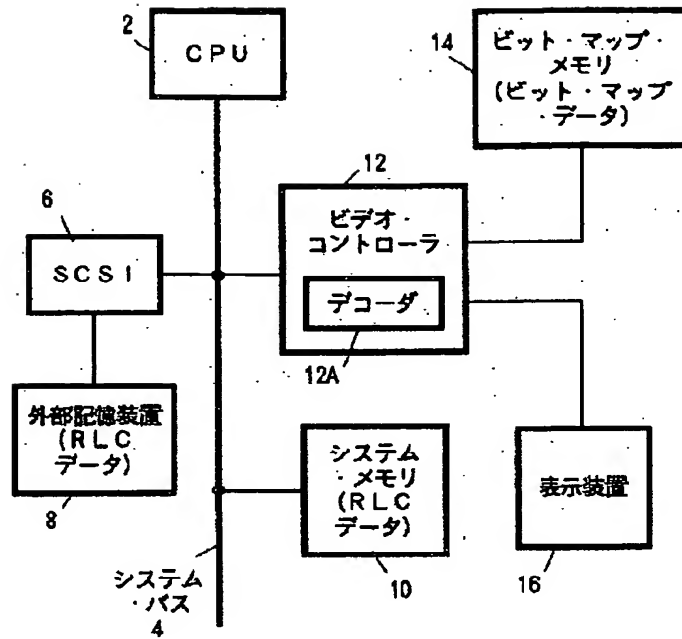
【図11】前記物理アドレス発生器のビット・マップ・メモリ・アドレス発生器を示すブロック図である。

【図12】RLCデコーダ中の物理アドレス発生器とRLCオペレーション・サブ・コントローラとの間のフィードバック・ループを示すブロック図である。

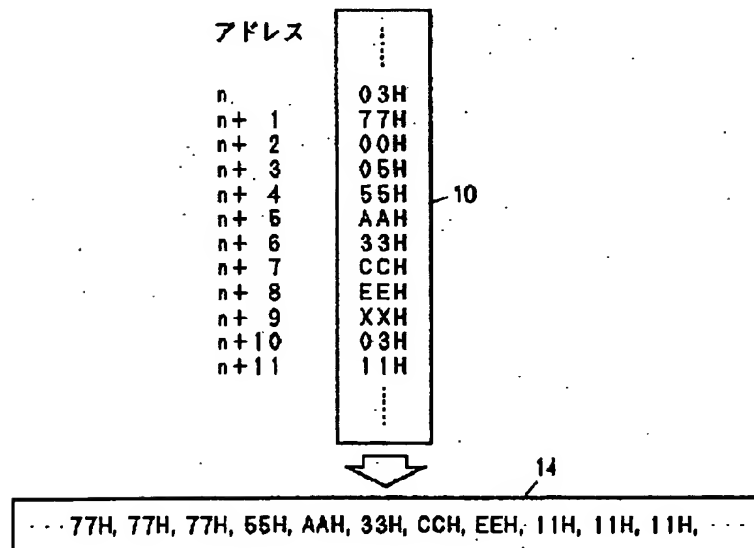
【符号の説明】

2	プロセッサ (CPU)
4	システム・バス
8	外部記憶装置
10	システム・メモリ
12	ビデオ・コントローラ
12A	デコーダ
14	ビット・マップ・メモリ
16	表示装置
18	I/Oレジスタ群
19	動作開始命令レジスタ
20	RLCオペレーション・コントローラ
22	RLCオペレーション・コントロール・カウンタ
24	次ステート発生器
26	コマンド発生器
27	コマンド・タイミング発生器
28	RLCオペレーション・サブ・コントローラ
29	ハンドシェイク信号タイミング調整器
30	RLCグラフィック・データ・コントローラ
32	同期フィードバック・ループ
34	同期フィードバック・ループ
36	非同期フィードバック・ループ
38	非同期回路
40	物理アドレス発生器
40A	システム・メモリ・アドレス発生器
40B	ビット・マップ・メモリ・アドレス発生器
62	システム・メモリ・アドレス・カウンタ
66	ビット・マップ・メモリ・アドレス・カウンタ
82	リード・スタート・アドレス・レジスタ
84	ライト・スタート・アドレス・レジスタ

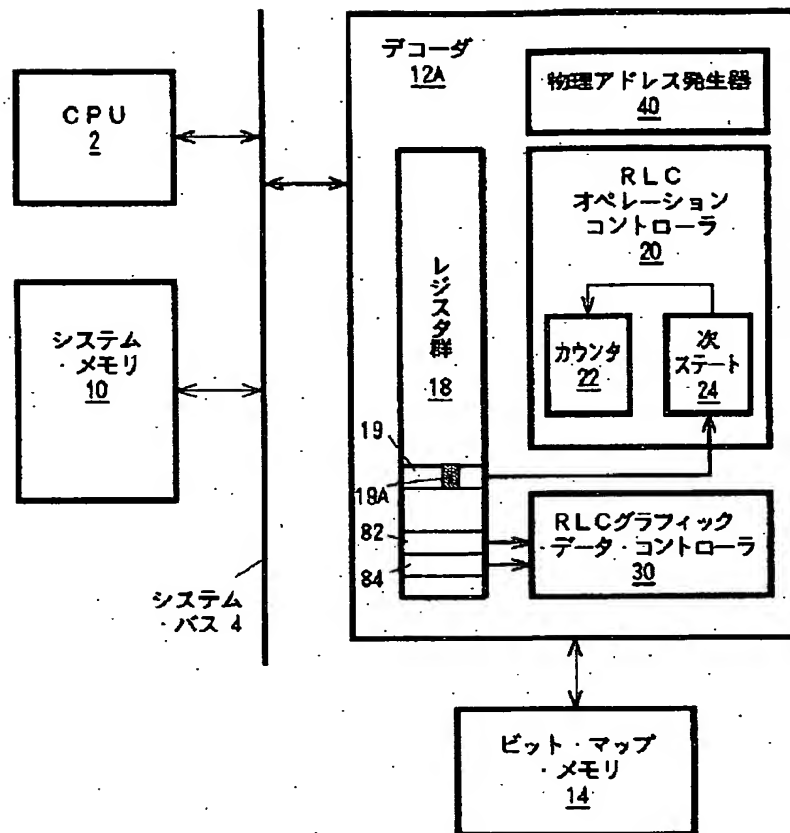
【図1】



【図2】



【図 3】



【図 8】

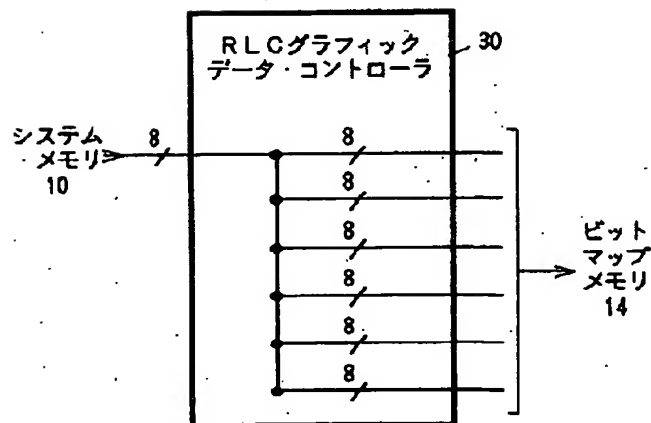


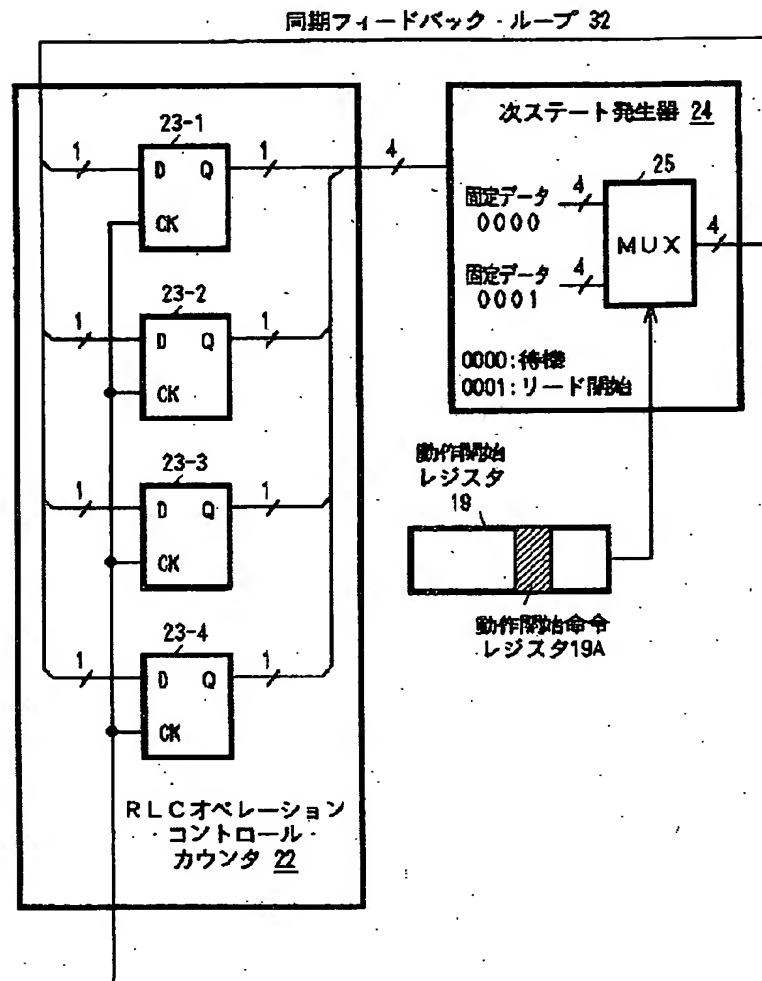
Figure 1 is a block diagram of a PLL circuit. The circuit includes the following components and connections:

- Reference Signal (非同期フィードバックループ 36):** Provides an input to the **Handshake Signal Timing Adjuster (29)**.
- Handshake Signal Timing Adjuster (29):** Receives the reference signal and a **CK** signal. Its output is connected to the **Next State Generator (24)**.
- Next State Generator (24):** Receives the output from the timing adjuster and the **CK** signal. Its output is connected to the **RLC Operation Subcontroller (28)**.
- RLC Operation Subcontroller (28):** Receives the output from the next state generator and the **CK** signal. Its output is connected to the **Command Timing Adjuster (27)**.
- Command Timing Adjuster (27):** Receives the output from the RLC subcontroller and the **CK** signal. Its output is connected to the **RLC Operation Subcontroller (28)** and the **Feedback Signal (同期フィードバックループ 34)**.
- Feedback Signal (同期フィードバックループ 34):** Provides a feedback signal to the **Next State Generator (24)**.
- CK (Clock):** A common clock signal provided to the timing adjusters and the next state generator.
- Output (同期回路 38):** The output of the feedback signal is connected to the **同期回路 (Synchronous Circuit) 38**.

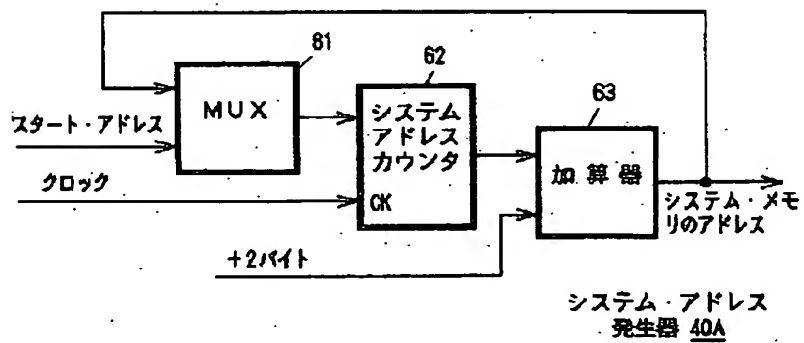
```

graph LR
    A[システムメモリ 10] -- n --> B[ラッチ 52]
    B -- n --> C[データ・ローデータ 54]
    C -- n --> D[データ抽換器 56]
    D -- n --> E[ビットマップメモリ 14]
    F[RLCグラフィックデータ・コントローラ 30]
  
```

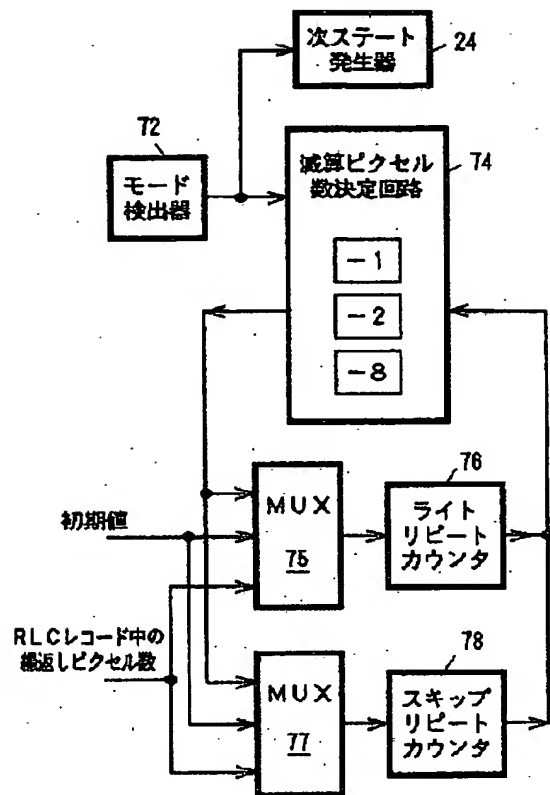

【図5】



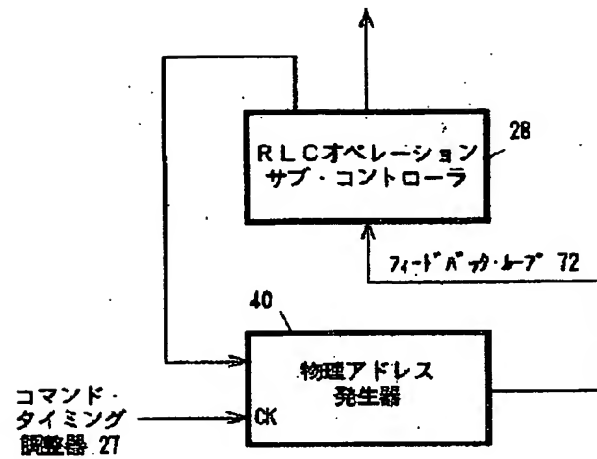
【図10】



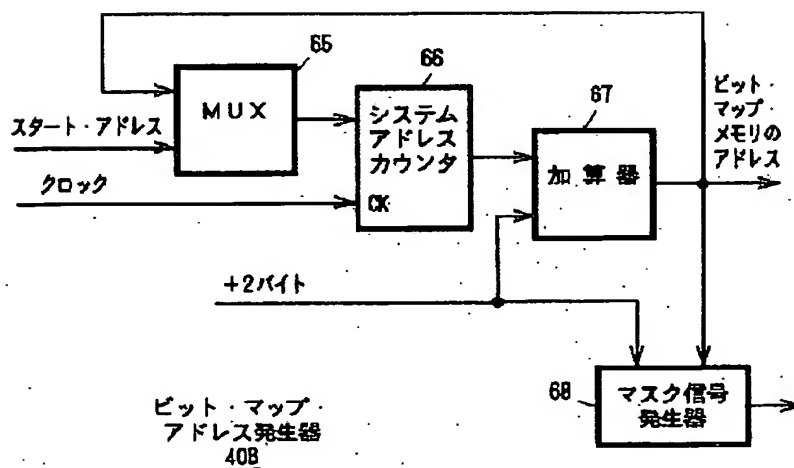
【図6】



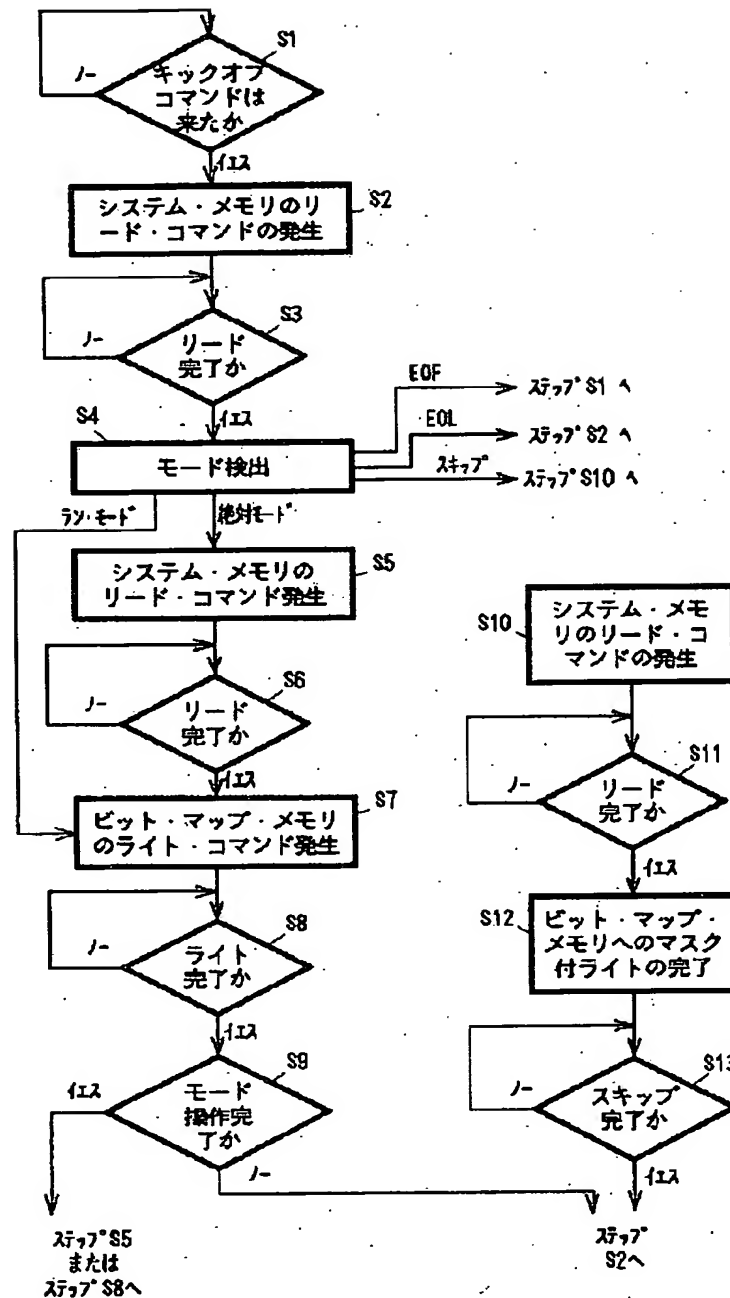
【図12】



【図11】



【図7】



フロントページの続き

(72)発明者 合田 裕二
 神奈川県大和市下鶴間1623番地14 日本ア
 イ・ビー・エム株式会社 大和事業所内

(72)発明者 ダーウィン・ピー・ラックレイ
 アメリカ合衆国フロリダ州ボカ・ラトン
 エンフィールド・ストリート756番地